

REMARKS/ARGUMENTS

1. Claims 25-36 Comply with 35 U.S.C. §101

The Examiner found that the “article of manufacture” claims 25-36 are not directed to a medium and thus not statutory as required under 35 U.S.C. §101. (Office Action, pg. 2) Applicants traverse.

The Application expressly defines an “article of manufacture” as “used herein refers to code or logic implemented in a computer readable medium accessed and executed by a processor” or code “accessible through a transmission media or from a file server over a network.” (Application, pg. 35, para. 83) Thus, the claims are directed to code or logic implemented in a tangible medium (i.e., the computer readable medium or transmission media) that causes operations to be performed as recited in the limitations.

The Manual of Patent Examination and Procedure (MPEP) states that “a claimed computer-readable medium encoded with a computer program is a computer element which defines structural and functional interrelationships between the computer program and the rest of the computer which permit the computer program’s functionality to be realized, and is thus statutory.” MPEP Sec. 2106(a), pg. 2100-13. Here, the claims are directed to an article of manufacture expressly defined as code or logic implemented in a tangible medium (e.g., computer readable medium, transmission media, etc.) that defines structural and functional interrelationships which permit the code and logic functionality to be realized. All the limitations comprise the functionality implemented in the article of manufacture

Accordingly, Applicants submit that the “article of manufacture claims” are directed to statutory subject matter and comply with 35 U.S.C. §101 and request that this rejection be withdrawn.

2. Claims 1-36 are Patentable Over the Cited Art

The Examiner rejected claims 1-36 as obvious (35 U.S.C. §103) as obvious over Graylin (U.S. App. Pub. No. 2003/0033415) and Rajarajan (U.S. App. Pub. No. 2002/00143949). Applicants traverse.

Claims 1, 13, and 25 concern enabling access to resource objects in an application engine, and require: receiving a request, from a calling entity, for resource objects of a specified type in the application engine; generating a request to the application engine for information on available resource objects of the specified type; in response to receiving the information from the

application engine, generating a collection object including one metadata element for each resource object of the specified type in the application engine; and returning the generated collection object to the calling entity.

The Examiner cited pg. 2, para. 33 of Graylin as teaching the claim requirements of receiving a request, from a calling entity, for resource objects of a specified type in the application engine and generating a request to the application engine for information on available resource objects of the specified type. (Office Action, pg. 2) Applicants traverse.

The cited pg. 2 of Graylin discusses a user preference elaborating system including an entitlement processor which receives data from and provides data to an accessor data storage, accessor group data storage and an object registry. The accessors are entities request access to objects or resources. An accessor group refers to a collection of accessors and an object registry includes individual resources associated with an entitlement expression. An entitlement expression is a specification of access entitlement and has a reference to at least one accessor group and one or more operators.

Paragraph 45, pg. 4 of Graylin further mentions that the entitlement processor receives an entitlement verification request from a client processing wishing to access a resource that includes an accessor name and an object name or ID indicating the object the client process wishes to access.

Nowhere does the above cited Graylin anywhere disclose that the entitlement processor receive a request from a calling entity, i.e., one client, and then generate a request to an application engine for information on available resource objects of the specified type as claimed. In fact, Graylin teaches away from this requirement because Graylin mentions that the entitlement processor queries an accessor table to determine an accessor ID (see, pg. 4, para. 45). Querying the accessor ID as mentioned in Graylin does not teach or suggest the claim requirement of generating a request for information on available resource objects of a specified type. Graylin further mentions querying an object registry to retrieve an object's e-expression that includes accessor names that are allowed to access a resource. (pg. 4, para. 48). Querying an object registry to determine accessors that can access a resource does not teach or suggest the claim requirement of generating a request for information on available resource objects of a specified type.

Moreover, Applicants submit that Graylin additionally teaches away from the claim requirement of generating a request for information on available resource objects of the specified

type from the calling entity because the client process request to the entitlement manager of Graylin identifies the object to access. (See, pg. 4, para. 45) Thus, there is no need in Graylin to generate a request for information on available resource objects of a specified type because the request to the entitlement manager in Graylin already specifies the object to access.

Thus, the cited Graylin does not teach or suggest the generating the request limitation for which it is cited.

The Examiner cited pg. 2, para. 11 and pg. 12, para. 101 of Rajarajan as teaching the claim requirements of in response to receiving the information from the application engine, generating a collection object including one metadata element for each resource object of the specified type in the application engine and returning the generated collection object to the calling entity. (Office Action, pg. 3) Applicants traverse.

The cited pg. 2 of Rajarajan mentions maintaining a plurality of resources in a task based manner. A method receives information from resources related to tasks associated with a same type of object and stores the information from the first resource in association with the second resource. The method further receives a request to perform the management task in relation to the first managed object and determines which resource to call in response to the request.

The cited pg. 2 discusses how to store information from resources related to objects and to perform a management task with respect to a managed object. Nowhere does this cited pg. 2 anywhere teach, suggest or mention the claim requirements of in response to receiving the information from the application engine, generating a collection object including one metadata element for each resource object of the specified type in the application engine that is returned to a calling entity requesting resource objects of a specified type. There is no teaching of a collection object as claimed in the cited pg. 2.

The cited pg. 12 of Rajarajan mentions that a task handler address is used to generate a request that is sent to the identified resource to collect all dynamic tasks. Task information relates to functions that may be performed on a particular data object, but may not be available for objects of that type. A dynamic task may relate to a particular instance of an object. Para. 102 mentions that the dynamic tasks may be received and merged to form a task list.

Although the cited pg. 12 discusses how to collect information on tasks relating to functions performed on a particular object, nowhere does the cited pg. 12 anywhere teach or suggest generating a collection object including one metadata element for each resource object of a specified type. Applicants submit collecting information on tasks relating to functions

performed on a particular object as mentioned in the cited Rajarajan does not teach or suggest the claim requirement of a collection object including a metadata element for each resource object of a specified type in an application engine.

Thus, the cited Rajarajan does not teach or suggest the limitations for which it is cited.

Even if one were to combine Graylin and Rajarajan as the Examiner proposes, the cited combination still does not teach or suggest the claim requirements for the reasons discussed above.

Accordingly, claims 1, 13, and 25 are patentable over the cited art because the cited combination does not teach or suggest all the claim requirements.

Claims 2-12, 14-24 and 26-36 are patentable over the cited art because they depend from one of claims 1, 13, and 25, which are patentable over the cited art for the reasons discussed above. Moreover, the below discussed dependent claims provide additional grounds of patentability over the cited art.

Claims 4, 16, and 28 depend from claims 1, 13, and 25, respectively, and further require that the application engine is one of a plurality of service engines enabling access to service resources, wherein the request for the resource objects from the calling entity comprises a method that is a member of a service class implementation of the application engine, wherein each service engine provides one service class implementation of methods and objects from a same abstract service class. The Examiner cited pg. 7, paras. 72-74 of Rajarajan as teaching the additional requirements of these claims. (Office Action, pg. 4) Applicants traverse.

The cited para. 72 mentions that a configuration manager handles the addition of new resources and communicates with the resources and may configure the resources to allow management of those resources. The configuration manager also provides other managers information on a registered resource. The configuration manager is a web service for which web service methods are provided and other managers may use the methods to get information about the resources.

Nowhere does the above cited pg. 7 anywhere teach or suggest that the cited configuration manager is one of a plurality of service engines enabling access to service resources. Further, nowhere does the cited pg. 7 anywhere teach or suggest the claim requirement of multiple service engines, each providing one service class implementation of methods and objects from a same abstract service class. In fact, the cited pg. 7 teaches away

from this requirement because pg. 7 and FIG. 3 shows only one configuration manager 330, not multiple service engines each implementing a same abstract service class as claimed.

Accordingly, claims 4, 16, and 28 provide additional grounds of patentability over the cited art.

Claims 5, 17, and 29 depend from claims 1, 13, and 25 and further require that the application engine and other service engines comprise workflow products from different vendors. The Examiner cited pg. 20, para. 175 of Rajarajan as teaching the additional requirements of these claims. (Office Action, pg. 4) Applicants traverse.

The cited pg. 20 discusses a search driven model for locating and working with objects without having to navigate through varying applications. The system provides a framework that allows an administrator to work with a specific object or group of objects. Once an object is located, the user may perform tasks associated with that object.

Nowhere does the cited pg. 20 anywhere teach or suggest multiple application or service engines comprising workflow products from different vendors. There is no mention of products from different vendors. Instead, the cited pg. 20 discusses a framework to allow an administrator to search and work with objects.

Accordingly, claims 5, 17, and 29 provide additional grounds of patentability over the cited art.

Claims 6, 18, and 30 depend from claims 5, 17, and 29 and further require that the workflow service class implementations from different vendors each include methods and objects from a same abstract workflow service class specifying methods and objects to include in all workflow service class implementations. The Examiner cited pg. 8, para. 95 of Graylin as teaching the additional requirements of these claims. (Office Action, pg. 5)

The cited pg. 8 of Graylin mentions a distributed software environment based on middleware, which is connectivity software including a set of enabling services that allow multiple processes running on one or machines to interact, such as CORBA and COM/DCOM.

Nowhere does this cited pg. 8 of Graylin teach or suggest a workflow service class implementations from different vendors of a same abstract workflow service class. There is no mention of a workflow service class in the cited pg. 8 nor workflow service class implementations from different vendors. Instead, the cited pg. 8 discusses middleware.

Accordingly, claims 6, 18, and 30 provide additional grounds of patentability over the cited art.

Claims 7, 19, and 31 depend from claims 1, 13, and 25 and further require that the application engine comprises a workflow engine and that the specified type of the requested resource objects comprises one of workflow objects, workflow templates and work lists defined in the application engine. The Examiner cited pg. 8, para. 91 of Graylin as teaching the additional requirements of these claims. (Office Action, pg. 5) Applicants traverse.

The cited para. 91 mentions that a client comprises any object that uses the resources of another object, such as a server object. The server objects can be accessed by client objects seeking user preference information by the invocation of preference manager methods.

Nowhere does this cited para. 91 anywhere teach or suggest that the application engine is a workflow engine and the requested resource objects comprise workflow objects, workflow templates and work lists defined in the application engine. Nowhere in the cited para. 91 is there any teaching or mention of workflow objects as claimed.

Accordingly, claims 7, 19, and 31 provide additional grounds of patentability over the cited art.

The additional dependent claims 8-12, 20-24, and 32-36 provide additional requirements that in combination with the base and dependent claims provide further grounds of patentability over the cited art.

Conclusion

For all the above reasons, Applicant submits that the pending claims 1-36 are patentable over the art of record. Applicants have not added any claims. Nonetheless, should any additional fees be required, please charge Deposit Account No. 09-0460.

//

//

//

//

//

//

//

//

The attorney of record invites the Examiner to contact him at (310) 553-7977 if the Examiner believes such contact would advance the prosecution of the case.

Dated: February 28, 2005

By. 

David W. Victor
Registration No. 39,867

Please direct all correspondences to:

David Victor
Konrad Raynes & Victor, LLP
315 South Beverly Drive, Ste. 210
Beverly Hills, CA 90212
Tel: 310-553-7977
Fax: 310-556-7984